Lógica
Aula 18

Leliane Nunes de Barros

2018

leliane@ime.usp.br

Fases da verificação (recordação)

1. Partindo de uma descrição informal R, gerar φ_R.
2. Escrever programa P.
3. Provar que $P \vdash \varphi_R$.

Triplas de Hoare (recordação)

Uma **especificação**, dada pela tripla de Hoare

$$(|\varphi|)P(|\psi|)$$

é válida, se ao rodar o programa P num estado que satisfaz φ, leva a um estado que satisfaz ψ.

- φ e ψ são fórmulas da LPO no universo da aritmética inteira.
- P é um programa escrito numa linguagem imperativa.

Correção Total e Parcial (recordação)

$$\vDash_{tot} (|\varphi|)P(|\psi|)$$

$$\Updownarrow$$

$\vDash_{par} (|\varphi|)P(|\psi|)$ e P termina.

Verificação de Programas: Regras de Demonstração

$$\frac{(|\varphi|)\, C_1\, (|\eta|) \qquad (|\eta|)\, C_2\, (|\psi|)}{(|\varphi|)\, C_1; C_2\, (|\psi|)}\ \textit{composição}$$

Verificação de Programas: Regras de Demonstração

$$\frac{(|\varphi|)\, C_1\, (|\eta|) \qquad (|\eta|)\, C_2\, (|\psi|)}{(|\varphi|)\, C_1; C_2\, (|\psi|)}\ \textit{composição}$$

$$\frac{}{(|\psi[E/x]|)\, x = E\, (|\psi|)}\ \textit{atribuição}$$

Verificação de Programas: Regras de Demonstração

$$\frac{(|\varphi|)\,C_1\,(|\eta|) \qquad (|\eta|)\,C_2\,(|\psi|)}{(|\varphi|)\,C_1; C_2\,(|\psi|)}\ \textit{composição}$$

$$\frac{}{(|\psi[E/x]|)\,x = E\,(|\psi|)}\ \textit{atribuição}$$

$$\frac{\vdash_{AR} \varphi' \to \varphi \qquad (|\varphi|)\,C\,(|\psi|) \qquad \vdash_{AR} \psi \to \psi'}{(|\varphi'|)\,C\,(|\psi'|)}\ \textit{implicação}$$

Escrevendo uma prova sequencial (tabela de prova)

$$\vdash_{par} (|\varphi_0|) P (|\varphi_n|)$$

Sendo $P = C_1, C_2, ..., C_n$ disposto numa tabela de prova:

$C_1;$

$C_2;$

$\vdots$

$C_n;$

Escrevendo uma prova sequencial (tabela de prova)

$$\vdash_{par} (|\varphi_0|) P (|\varphi_n|)$$

Sendo $P = C_1, C_2, ..., C_n$ disposto numa tabela de prova:

$C_1;$

$C_2;$

$\vdots$

$C_n;$

$(|\varphi_n|)$

Escrevendo uma prova sequencial (tabela de prova)

$$\vdash_{par} (|\varphi_0|) P (|\varphi_n|)$$

Sendo $P = C_1, C_2, ..., C_n$ disposto numa tabela de prova:

$C_1;$

$C_2;$

$\vdots$

$(|\varphi_{n-1}|)$

$C_n;$

$(|\varphi_n|)$

Escrevendo uma prova sequencial (tabela de prova)

$$\vdash_{par} (|\varphi_0|) P (|\varphi_n|)$$

Sendo $P = C_1, C_2, ..., C_n$ disposto numa tabela de prova:

$C_1;$

$C_2;$

$(|\varphi_2|)$

$\vdots$

$(|\varphi_{n-1}|)$

$C_n;$

$(|\varphi_n|)$

Escrevendo uma prova sequencial (tabela de prova)

$$\vdash_{par} (|\varphi_0|) P (|\varphi_n|)$$

Sendo $P = C_1, C_2, ..., C_n$ disposto numa tabela de prova:

$$
\begin{array}{l}
C_1; \\
\quad (|\varphi_1|) \\
C_2; \\
\quad (|\varphi_2|) \\
\vdots \\
\quad (|\varphi_{n-1}|) \\
C_n; \\
\quad (|\varphi_n|)
\end{array}
$$

Escrevendo uma prova sequencial (tabela de prova)

$$\vdash_{par} (|\varphi_0|) P (|\varphi_n|)$$

Sendo $P = C_1, C_2, ..., C_n$ disposto numa tabela de prova:

$$
\begin{array}{l}
\quad (|\varphi_0|) \\
C_1; \\
\quad (|\varphi_1|) \\
C_2; \\
\quad (|\varphi_2|) \\
\vdots \\
\quad (|\varphi_{n-1}|) \\
C_n; \\
\quad (|\varphi_n|)
\end{array}
$$

Escrevendo uma prova sequencial

- Cada φ_i deve ser válida no ponto em que aparece no programa.

Escrevendo uma prova sequencial

- Cada φ_i deve ser válida no ponto em que aparece no programa.

- Cada transição

$$(|\varphi_{i-1}|)$$

$$C_i;$$

$$(|\varphi_i|)$$

usa alguma regra do cálculo e parte de φ_i para calcular a *pré-condição mais fraca* φ_{i-1}.

Uso da Regra da Implicação na Prova Sequencial

$$\frac{\vdash_{AR} \varphi' \to \varphi \qquad (|\varphi|)\,C\,(|\psi|) \qquad \vdash_{AR} \psi \to \psi'}{(|\varphi'|)\,C\,(|\psi'|)}\ \textit{implicação}$$

Uso da Regra da Implicação na Prova Sequencial

$$\frac{\vdash_{AR} \varphi' \to \varphi \qquad (|\varphi|)\,C\,(|\psi|) \qquad \vdash_{AR} \psi \to \psi'}{(|\varphi'|)\,C\,(|\psi'|)}\ \textit{implicação}$$

Essa regra permite escrever (usando a Dedução Natural) a prova sequencial:

$$(|\varphi|)$$

$$(|\varphi'|)$$

sem ter um comando intercalado, quando $\vdash_{AR} \varphi \to \varphi'$.

Exemplos

$$\vdash_{par} (|y < 3|) y = y + 1 (|y < 4|)$$

```
y = y + 1;
```

Exemplos

$$\vdash_{par} (\!|y < 3|\!) y = y + 1 (\!|y < 4|\!)$$

```
y = y + 1;
```

$(\!|y < 4|\!)$

Exemplos

$$\vdash_{par} (|y < 3|) y = y + 1 (|y < 4|)$$

$(|y + 1 < 4|)$

```
y = y + 1;
```

$(|y < 4|)$ Atribuição

Exemplos

$$\vdash_{par} (|y < 3|) y = y + 1 (|y < 4|)$$

$(|y < 3|)$

$(|y + 1 < 4|)$ Implicação

```
y = y + 1;
```

$(|y < 4|)$ Atribuição

Exemplos

$$\vdash_{par} (|\top|) P (|z = x + y|)$$

```
z = x;

z = z + y;
```

Exemplos

$$\vdash_{par} (|\top|) P (|z = x + y|)$$

```
z = x;

z = z + y;
```

$$(|z = x + y|)$$

Exemplos

$$\vdash_{par} (|\top|) P (|z = x + y|)$$

```
z = x;
```

$(|z + y = x + y|)$

```
z = z + y;
```

$(|z = x + y|)$ Atribuição

Exemplos

$$\vdash_{par} (|\top|)P(|z = x + y|)$$

$(|x + y = x + y|)$

```
z = x;
```

$(|z + y = x + y|)$ Atribuição

```
z = z + y;
```

$(|z = x + y|)$ Atribuição

Exemplos

$$\vdash_{par} (\!|\top|\!) P (\!|z = x + y|\!)$$

$(\!|\top|\!)$

$(\!|x + y = x + y|\!)$ Implicação

```
z = x;
```

$(\!|z + y = x + y|\!)$ Atribuição

```
z = z + y;
```

$(\!|z = x + y|\!)$ Atribuição

Regra de demonstração do *if*

Regra do *if*

$$\frac{(|\varphi \wedge B|)\, C_1\, (|\psi|) \qquad (|\varphi \wedge \neg B|)\, C_2\, (|\psi|)}{(|(\varphi)|)\, \texttt{if}\ \ B\ \{C_1\}\ \texttt{else}\ \ \{C_2\}(|\psi|)}\ \mathit{if}$$

Regra de demonstração do *if*

Regra do *if*

$$\frac{(|\varphi \wedge B|)\, C_1\, (|\psi|) \qquad (|\varphi \wedge \neg B|)\, C_2\, (|\psi|)}{(|(\varphi)|)\, \texttt{if}\ \ B\, \{C_1\}\, \texttt{else}\ \ \{C_2\}(|\psi|)}\ \mathit{if}$$

- ramifica em B e $\neg B$
- para ser usada numa demosntração precisa ser modificada

Regra de demonstração do *if* modificada

Regra do *if* modificado

$$\frac{(|\varphi_1|)\, C_1\, (|\psi|) \qquad (|\varphi_2|)\, C_2\, (|\psi|)}{(|(B \to \varphi_1) \land (\neg B \to \varphi_2)|)\,\texttt{if}\ B\,\{C_1\}\,\texttt{else}\ \{C_2\}(|\psi|)}\ \mathit{if}'$$

Uso numa demonstração:

- em cada ramificação B e $\neg B$, adicionamos ψ no final de cada bloco C_1 e C_2, e em seguida "subimos" fazendo demonstrações para encontrar φ_1 e φ_2 no início de cada bloco e
- finalmente, inserimos $(|(B \to \varphi_1) \land (\neg B \to \varphi_2)|)$ imediatamente antes do comando *if* $B\ \{C_1\}$ *else* $\{C_2\}$

Exemplo

$(|\top|)$

```
a = x + 1;
if (a - 1 == 0) {
  y = 1;
} else {
  y = a;
}
```

$(|y = x + 1|)$

Exemplo

$(|\top|)$

```
a = x + 1;

if (a - 1 == 0) {

  y = 1;
```

$(|y = x + 1|)$

```
} else {

  y = a;
```

$(|y = x + 1|)$

```
}
```

$(|y = x + 1|)$ If'

Exemplo

$(|\top|)$

```
a = x + 1;
```

```
if (a - 1 == 0) {
```

```
  y = 1;
```

$(|y = x + 1|)$

```
} else {
```

$(|a = x + 1|)$

```
  y = a;
```

$(|y = x + 1|)$ Atribuição

```
}
```

$(|y = x + 1|)$ If'

Exemplo

$(\!|\top|\!)$

```
a = x + 1;

if (a - 1 == 0) {

  y = 1;
```

$(\!|y = x + 1|\!)$

```
} else {
```

$(\!|a = x + 1|\!)$ If' (φ_2)

```
  y = a;
```

$(\!|y = x + 1|\!)$ Atribuição

```
}
```

$(\!|y = x + 1|\!)$ If'

Exemplo

$(\!|\top|\!)$

```
a = x + 1;
```

```
if (a - 1 == 0) {
```

$(\!|1 = x + 1|\!)$

```
  y = 1;
```

$(\!|y = x + 1|\!)$ Atribuição

```
} else {
```

$(\!|a = x + 1|\!)$ If' (φ_2)

```
  y = a;
```

$(\!|y = x + 1|\!)$ Atribuição

```
}
```

$(\!|y = x + 1|\!)$ If'

Exemplo

$(\!|\top|\!)$

```
a = x + 1;
```

```
if (a - 1 == 0) {
```

$(\!|1 = x + 1|\!)$ If' (φ_1)

```
  y = 1;
```

$(\!|y = x + 1|\!)$ Atribuição

```
} else {
```

$(\!|a = x + 1|\!)$ If' (φ_2)

```
  y = a;
```

$(\!|y = x + 1|\!)$ Atribuição

```
}
```

$(\!|y = x + 1|\!)$ If'

Exemplo

$(\!|\top|\!)$

```
a = x + 1;
```

$(\!|((a-1=0) \to (1 = x+1)) \wedge$
$(\neg(a-1=0) \to (a = x+1))|\!)$

```
if (a - 1 == 0) {
```

$(\!|1 = x+1|\!)$ If' (φ_1)

```
  y = 1;
```

$(\!|y = x+1|\!)$ Atribuição

```
} else {
```

$(\!|a = x+1|\!)$ If' (φ_2)

```
  y = a;
```

$(\!|y = x+1|\!)$ Atribuição

```
}
```

$(\!|y = x+1|\!)$ If'

Exemplo

$(\!|\top|\!)$

$(\!|((x+1-1=0) \to (1=x+1))\wedge$
$(\neg(x+1-1=0) \to (x+1=x+1))|\!)$

```
a = x + 1;
```

$(\!|((a-1=0) \to (1=x+1))\wedge$
$(\neg(a-1=0) \to (a=x+1))|\!)$ Atribuição

```
if (a - 1 == 0) {
```

$(\!|1=x+1|\!)$ If' (φ_1)

```
  y = 1;
```

$(\!|y=x+1|\!)$ Atribuição

```
} else {
```

$(\!|a=x+1|\!)$ If' (φ_2)

```
  y = a;
```

$(\!|y=x+1|\!)$ Atribuição

```
}
```

$(\!|y=x+1|\!)$ If'

Exemplo

$(\!|\top|\!)$

$(\!|((x+1-1=0) \to (1=x+1))\wedge$
$(\neg(x+1-1=0) \to (x+1=x+1))|\!)$ Implicação

```
a = x + 1;
```

$(\!|((a-1=0) \to (1=x+1))\wedge$
$(\neg(a-1=0) \to (a=x+1))|\!)$ Atribuição

```
if (a - 1 == 0) {
```

$(\!|1=x+1|\!)$ If' (φ_1)

```
  y = 1;
```

$(\!|y=x+1|\!)$ Atribuição

```
} else {
```

$(\!|a=x+1|\!)$ If' (φ_2)

```
  y = a;
```

$(\!|y=x+1|\!)$ Atribuição

```
}
```

$(\!|y=x+1|\!)$ If'

Regra de demonstração do *while*

Regra do *While*

$$\frac{(|\eta \wedge B|)\,C\,(|\eta|)}{(|\eta|)\texttt{while}\;\; B\,\{C\}\,(|\eta \wedge \neg B|)}\;\textit{while}$$

Regra de demonstração do *while*

Regra do *While*

$$\frac{(|\eta \wedge B|)\, C\, (|\eta|)}{(|\eta|)\texttt{while } B\ \{C\}\ (|\eta \wedge \neg B|)}\ \textit{while}$$

Uso numa prova sequencial:

$(|\eta|)$

`while B`

$\qquad (|\eta \wedge B|)$

$\qquad$ `C`

$\qquad (|\eta|)$

$(|\eta \wedge \neg B|)$

Regra de demonstração do *while*

Normalmente, queremos provar

$$(|(\varphi)|)\texttt{while}\ B\,\{C\}\,(|\psi|)$$

Regra de demonstração do *while*

Normalmente, queremos provar

$$(|(\varphi)|)\texttt{while } B\,\{C\}\,(|\psi|)$$

Mas, vamos achar η (invariante) tal que:

1. $\vdash_{AR} \varphi \to \eta$

Regra de demonstração do *while*

Normalmente, queremos provar

$$(|(\varphi)|)\texttt{while } B\,\{C\}\,(|\psi|)$$

Mas, vamos achar η (invariante) tal que:

1. $\vdash_{AR} \varphi \to \eta$
2. $\vdash_{AR} \eta \land \neg B \to \psi$

Regra de demonstração do *while*

Normalmente, queremos provar

$$(|(\varphi)|)\texttt{while } B\,\{C\}\,(|\psi|)$$

Mas, vamos achar η (invariante) tal que:

1. $\vdash_{AR} \varphi \to \eta$
2. $\vdash_{AR} \eta \wedge \neg B \to \psi$
3. $\vdash_{par} (|(\eta)|)\texttt{while } B\,\{C\}\,(|\eta \wedge \neg B|)$

Invariante

Um *invariante* para um laço *while* B $\{C\}$ é uma propriedade η que deve ser satisfeita antes e no final de cada iteração do *while*, isto é:

$$\vDash_{par} (|\eta \wedge B|) C (|\eta|)$$

Invariante

Um *invariante* para um laço *while* B $\{$C$\}$ é uma propriedade η que deve ser satisfeita antes e no final de cada iteração do *while*, isto é:

$$\vDash_{par} (|\eta \wedge B|)\, C\, (|\eta|)$$

Obs.: η pode ser falso **durante** a execução de C.

Invariante

Um *invariante* para um laço *while* B $\{$C$\}$ é uma propriedade η que deve ser satisfeita antes e no final de cada iteração do *while*, isto é:

$$\vDash_{par} (|\eta \wedge B|)\,C\,(|\eta|)$$

Obs.: η pode ser falso **durante** a execução de C.

Exemplos

Fatorial:

```
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
```

$$\vdash_{par} (|\top|) \text{ Fatorial } (|y = x!|)$$

Para aplicar a regra de demonstração somente sobre o *while*, consideramos a pré-condição $(|y = 1 \wedge z = 0|)$, isto é

$$\vdash_{par} (|y = 1 \wedge z = 0|) \text{ While B } \{ \text{ C } \} \; (|y = x!|)$$

Exemplos

Potência:

```
y = 1;
z = 0;
while (z != n) {
   z = z + 1;
   y = y * x;
}
```

$$\vdash_{par} (|\top|)\ \text{Potência}\ (|y = x^n|)$$

Aplicando a regra do *while*

1. Adivinhar η

Aplicando a regra do *while*

1. Adivinhar η
2. Provar $\vdash_{AR} \phi \rightarrow \eta$ e $\vdash_{AR} \eta \wedge \neg B \rightarrow \psi$ (se falhar, volta para o primeiro passo)

Aplicando a regra do *while*

1. Adivinhar η
2. Provar $\vdash_{AR} \phi \to \eta$ e $\vdash_{AR} \eta \land \neg B \to \psi$ (se falhar, volta para o primeiro passo)
3. Subir η por C, obtendo η'

Aplicando a regra do *while*

1. Adivinhar η
2. Provar $\vdash_{AR} \phi \to \eta$ e $\vdash_{AR} \eta \wedge \neg B \to \psi$ (se falhar, volta para o primeiro passo)
3. Subir η por C, obtendo η'
4. Provar que $\vdash_{AR} \eta \wedge B \to \eta'$: η é invariante (se falhar, volta para o primeiro passo)

Aplicando a regra do *while*

1. Adivinhar η
2. Provar $\vdash_{AR} \phi \rightarrow \eta$ e $\vdash_{AR} \eta \wedge \neg B \rightarrow \psi$ (se falhar, volta para o primeiro passo)
3. Subir η por C, obtendo η'
4. Provar que $\vdash_{AR} \eta \wedge B \rightarrow \eta'$: η é invariante (se falhar, volta para o primeiro passo)
5. Colocar η acima do while e φ acima do η.

Regras de Demonstração: resumo para correção parcial

$$\frac{(|\varphi|)\,C_1\,(|\eta|) \qquad (|\eta|)\,C_2\,(|\psi|)}{(|\varphi|)\,C_1;\,C_2\,(|\psi|)}\ \textit{composição}$$

$$\frac{}{(|\psi[E/x]|)\,x = E\,(|\psi|)}\ \textit{atribuição}$$

$$\frac{\vdash_{AR} \varphi' \to \varphi \qquad (|\varphi|)\,C\,(|\psi|) \qquad \vdash_{AR} \psi \to \psi'}{(|\varphi'|)\,C\,(|\psi'|)}\ \textit{implicação}$$

$$\frac{(|\varphi_1|)\,C_1\,(|\psi|) \qquad (|\varphi_2|)\,C_2\,(|\psi|)}{(|(B \to \varphi_1) \wedge (\neg B \to \varphi_2)|)\texttt{if }\ B\,\{C_1\}\ \texttt{else }\ \{C_2\}(|\psi|)}\ \textit{if}'$$

$$\frac{(|\eta \wedge B|)\,C\,(|\eta|)}{(|\eta|)\texttt{while }\ B\,\{C\}\ (|\eta \wedge \neg B|)}\ \textit{while}$$

sendo η um invariante do *while*. Uso no caso geral (com implicação):

$$\frac{\vdash_{AR} \varphi \to \eta \qquad \vdash_{par} (|(\eta)|)\texttt{while }\ B\,\{C\}\ (|\eta \wedge \neg B|) \qquad \vdash_{AR} \eta \wedge \neg B \to \psi}{(|(\varphi)|)\texttt{while }\ B\,\{C\}\ (|\psi|)}\ \textit{implica}_{\textit{while}}$$